In the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application.

Claim 1 (Previously presented): A method of copying a plurality of items, the method being implemented in a computer, the method comprising:

a parent process in said computer checking if a first item in said plurality is a file or a directory;

the parent process in said computer copying the first item to at least one storage media if the first item is found during the checking to be a file and the parent process creating a child process in said computer if the first item is found during the checking to be a directory;

if the first item is found during the checking to be a directory, the child process in said computer checking if another item in the directory represented by the first item is a file or a directory; and

if the first item is found during the checking to be a file, the parent process in said computer checking if a second item in said plurality is a file or a directory;

wherein at least one item in said plurality of items is copied to said at least one storage media.

Claims 2-3 (Canceled).

Claim 4 (Previously presented): The method of claim 1 further comprising, prior to the creating:

comparing a current number of processes, created for copying, with a limit; and

waiting if the current number is greater than or equal to the limit.

Claim 5 (Previously presented): The method of Claim 1 further comprising, prior to the creating:

the parent process increasing a limit on a resource; and

the child process using the resource at the increased limit during copying.

Claim 6 (Original): The method of claim 5 wherein:

the resource is number of open files.

Claim 7 (Original): The method of claim 5 wherein:

the resource is file size.

Claim 8 (Original): The method of claim 5 wherein:

the resource is memory.

Claim 9 (Original): The method of claim 8 wherein:

the memory is organized as a stack.

Claim 10 (Original): The method of claim 8 wherein:

the memory is organized as a heap.

Claim 11 (Previously presented): The method of claim 1 wherein the copying comprises:

transferring data from the file into a temporary buffer;

locking the temporary buffer; and

invoking a direct memory access (DMA) process for making a copy from the temporary buffer.

Claim 12 (Currently Amended): The method of claim [[11]] 1 further comprising prior to the copying:

the parent process checking if the first item is a link to itself, and performing said copying only if the first item is not a link to itself.

Claim 13 (Original): The method of claim 12 wherein:

the checking includes a string comparison operation.

Claim 14 (Currently Amended): The method of claim [[11]] 1 further comprising, during the copying:

the parent process sending an email message if a resource at a destination is full;

wherein the email message is sent to an email address of a user that started the method.

Claim 15 (Original): The method of claim 14 further comprising, during the copying:

waiting to be restarted subsequent to sending the email message.

Claim 16 (Original): The method of claim 15 wherein said waiting comprises:

sending a signal to self to suspend execution.

Claim 17 (Previously presented): The method of claim 15 further comprising, during the copying:

recopying said file from beginning, on being restarted.

Claim 18 (Previously presented): The method of claim 14 wherein:

the email address is identified from a password file based on an identity of said user.

Claim 19 (Previously presented): The method of claim 1 wherein:

said creating is performed only if said directory is not a current directory and not a parent directory.

Claims 20-28 (Canceled).

Claim 29 (Previously presented): A computer for copying a plurality of items, the computer comprising:

a processor comprising means for checking if an item to be copied in said plurality of items is a file or a directory; and

said processor further comprising means for copying the item to a storage media if the item is a file and creating a child process if the item is a directory;

wherein said child process comprises a copy of said means for checking and said means for copying and creating;

wherein when each item is input to said processor at least one item in said plurality of items is copied to said storage media.

Claim 30 (Previously presented): The computer of claim 29 further comprising:

means for sending an email message if the means for copying encounters an error.

Claim 31 (Previously presented): The computer of claim 29 further comprising:

means for increasing a limit on a resource to maximum.

Claim 32 (Previously presented): The apparatus of Claim 29

wherein said means for copying comprises:

means for transferring data from the file into a temporary buffer;

means for locking the temporary buffer; and

means for using direct memory access (DMA) to make a copy from the temporary buffer.

Claim 33 (Previously presented): The computer of claim 29 further comprising:

means for checking if the item is a link to itself.

Claim 34 (Previously presented): The method of Claim 1 wherein:

the parent process is started with an instruction to perform said method for each item in the directory.

Claim 35 (Canceled).

Claim 36 (Previously presented): The method of Claim 1 wherein:

the number of processes created corresponds to the number of directories to be copied.

Claim 37 (Canceled).

Claim 38 (Previously presented): The method of Claim 1 further comprising:

checking if the file is in a list of items to be excluded from copying; and

performing said copying only if the file is not in said list.

Claim 39 (Previously presented): The method of Claim 1 wherein:

the file is copied to multiple destinations if specified by the user.

Claims 40-42 (Canceled).

Claim 43 (Previously presented): A computer readable storage medium encoded with software, the software comprising instructions to archive an item in a computer by:

using a current process to check if said item is a file or a directory;

using the current process to copy the item to a storage media if the item is found during checking to be a file and creating a new process if the item is found during checking to be a directory; and

using the new process if created to check another item in the directory represented by said item;

wherein the new process if created executes simultaneously or contemporaneously with the current process; and

wherein at least one item is copied to at least said storage media.

Claim 44 (canceled).

Claim 45 (Previously presented): The medium of Claim 43 wherein:

the current process calls a function to recursively spawn a plurality of new processes including said new process; and

on return from the function, the current process waits for all new processes to finish.

Claim 46 (Currently Amended): ~~The method of Claim 1~~ A method of copying a plurality of items, the method being implemented in a computer, the method comprising:

a parent process in said computer checking if a first item in said plurality is a file or a directory;

the parent process in said computer copying the first item to at least one storage media if the first item is found during the checking to be a file and the parent process creating a child process in said computer if the first item is found during the checking to be a directory;

if the first item is found during the checking to be a directory, the child process in said computer checking if another item in the directory represented by the first item is a file or a directory; and

if the first item is found during the checking to be a file, the parent process in said computer checking if a second item in said plurality is a file or a directory;

wherein at least one item in said plurality of items is copied to said at least one storage media; and

wherein, if the first item is found during the checking to be a directory:

said child process in said computer is created after said parent process changes a default limit on a resource to a maximum limit;

after said creation, the child process inherits the maximum limit prior to performing the checking, the copying and the creating;

at least one of the parent process and the child process:

allocates memory to hold at least a temporary buffer and a stack, stores in said stack an absolute path and a local path to said directory, checks if an entry in said directory is a symbolic link, checks if the symbolic link is circular, and ignores the symbolic link if circular;

checks if a destination is full and sends an email message if a result thereof is true and waits to be restarted subsequent to said sending; and

transfers data to said temporary buffer, locks said temporary buffer and invokes a direct memory access process for making a copy from said temporary buffer to said destination.

Claim 47 (Previously presented): The method of Claim 46 further comprising:

using operating system stack space in said computer by making recursive calls when memory allocated for said stack is used up.

Claim 48 (Currently amended): The method of claim 1 further comprising:

the child process copying said another item to said at least one storage media if said another item is found to be a file and the child process creating another child process in said computer if said another item is found to be a directory; and

the parent process copying the second item to said at least one storage media if the second item is found to be a file and the parent process creating [[a]] yet another child process in said computer if the second item is found to be a directory.

Claim 49 (Previously presented): A computer readable storage medium comprising software that when executed causes a computer to perform a method, the software comprising:

a plurality of first instructions to check if a first item received as input is a file;

a plurality of second instructions, responsive to the first item being found to be a file, to copy the first item to at least one storage media;

a plurality of third instructions, responsive to the first item being found to be not a file and using as input a second item within a directory represented by said first item, to create a child process to execute said first instructions and to execute either said second instructions or said third instructions depending on whether said second item is a file;

wherein on execution of said software at least one of said first item and said second item is copied by said computer to at least said storage media.

Claim 50 (Previously presented): The computer readable storage medium of claim 49 further comprising:

a plurality of fourth instructions to check if the first item is a link to itself.

Claim 51 (Previously presented): The computer readable storage medium of claim 49 wherein:

the plurality of second instructions comprise instructions to check if said storage media is full and if so to send an email message.

Claim 52 (Previously presented): The computer readable storage medium of claim 49 further comprising:

a plurality of fourth instructions to be executed prior to creation of said child process, to compare a current number of child processes against a limit.

Claim 53 (Previously presented): The computer readable storage medium of claim 49 comprising:

a plurality of fourth instructions to be executed prior to creation of said child process to increase a limit on a resource.

Claim 54 (Previously presented): The computer readable storage medium of claim 49 comprising:

a plurality of fourth instructions to be executed prior to said first instructions, to check if an item in said plurality of items is in a list of items to be excluded from copying.

Claim 55 (New): The computer readable storage medium of claim 53 wherein:

the resource is number of open files.

Claim 56 (New): The computer readable storage medium of claim 53 wherein:

the resource is file size.

Claim 57 (New): The computer readable storage medium of claim 53 wherein:

the resource is memory.

Claim 58 (New): A computer readable storage medium comprising software that when executed causes a computer to copy a plurality of items, the software comprising:

instructions, for a parent process in said computer, to check if a first item in said plurality is a file or a directory;

instructions for the parent process in said computer to copy the first item to at least one storage media if the first item is found to be a file, and instructions for the parent process to create a child process in said computer if the first item is found to be a directory;

instructions to be executed if the first item is found to be a directory, for the child process in said computer to check if another item in the directory represented by the first item is a file or a directory; and

instructions to be executed if the first item is found to be a file, for the parent process in said computer to check if a second item in said plurality is a file or a directory;

wherein by execution of said instructions, at least one item in said plurality of items is copied to said at least one storage media; and

wherein during execution of said instructions, if the first item is found to be a directory:

said child process in said computer is created after said parent process changes a default limit on a resource to a maximum limit;

after said creation, the child process inherits the maximum limit prior to performing the checking, the copying and the creating;

at least one of the parent process and the child process:

allocates memory to hold at least a temporary buffer and a stack,

stores in said stack an absolute path and a local path to said directory, checks if an entry in said directory is a symbolic link, checks if the symbolic link is circular, and ignores the symbolic link if circular;

checks if a destination is full and sends an email message if a result thereof is true and waits to be restarted subsequent to said sending; and

transfers data to said temporary buffer, locks said temporary buffer and invokes a direct memory access process for making a copy from said temporary buffer to said destination.

Claim 59 (New): The computer readable storage medium of Claim 55 wherein the software further comprising:

instructions to use operating system stack space in said computer by making recursive calls when memory allocated for said stack is used up.